(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENT- UND MARKENAMT

(12) **Offenlegungsschrift**

(10) **DE 101 01 956 A 1**

(51) Int. Cl.[7]: **G 06 F 12/14**

(21) Aktenzeichen: 101 01 956.4
(22) Anmeldetag: 17. 1. 2001
(43) Offenlegungstag: 25. 7. 2002

DE 101 01 956 A 1

(71) Anmelder:
Infineon Technologies AG, 81669 München, DE

(74) Vertreter:
Epping, Hermann & Fischer, 80339 München

(72) Erfinder:
Klug, Franz, 81737 München, DE; Hartlieb, Heimo, Graz, AT; Sedlak, Holger, 85658 Egmating, DE

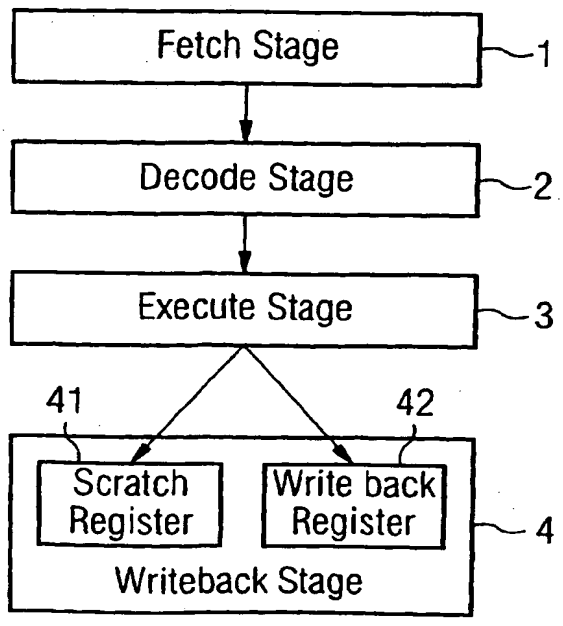
(56) Entgegenhaltungen:
DE 199 36 939 A1
WO 00 50 977 A1

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

(54) Verfahren zur Erhöhung der Sicherheit einer CPU

(57) Bei dem Verfahren wird eine Pipeline, bestehend aus einer Ladestufe (1), einer Decodierstufe (2), einer Ausführungsstufe (3) und einer Rückspeicherstufe (4), verwendet. Die Rückspeicherstufe besitzt mindestens ein Register (41), bei dessen Benutzung keine Zustandsänderung der CPU erfolgt, und mindestens ein Register (42), bei dessen Benutzung eine Zustandsänderung der CPU erfolgt. Erfindungsgemäß wird in der Decodierstufe mindestens eine zufällig ausgewählte Codesequenz als Platzhalter-Code oder Füllsel eingefügt, womit ein Angriff durch DPA erschwert wird.



DE 101 01 956 A 1

ICID: <DE_____10101956A1_I_>

Beschreibung

[0001] Die vorliegende Erfindung betrifft ein Verfahren zur Verbesserung der Sicherheit einer CPU.

[0002] Differential Power Analysis (DPA) ist ein bekanntes Angriffsszenario für Sicherheits-CPUs. Bei einem solchen Angriff wird eine Folge von Programmbefehlen und deren Auswirkungen in der CPU mittels statistischer Auswertungen der Kennlinien des Stromverbrauchs ermittelt. Aus diesen Auswertungen lassen sich detaillierte Rückschlüsse über das ausgeführte Programm gewinnen.

[0003] Aufgabe der vorliegenden Erfindung ist es, ein Verfahren zur Erhöhung der Sicherheit einer CPU anzugeben.

[0004] Diese Aufgabe wird mit dem Verfahren mit den Merkmalen des Anspruches 1 gelöst. Ausgestaltungen ergeben sich aus den abhängigen Ansprüchen.

[0005] Bei dem erfindungsgemäßen Verfahren wird eine als Pipeline aufgebaute CPU mit mindestens einer Decodierstufe und einer Rückspeicherstufe verwendet, die typisch eine Ladestufe (Fetch Stage), eine Decodierstufe (Decode Stage), eine Ausführungsstufe (Execute Stage) und eine Rückspeicherstufe (Writeback Stage) umfasst. Die Rückspeicherstufe besitzt mindestens ein Register, bei dessen Benutzung keine Zustandsänderung der CPU erfolgt, und mindestens ein Register, bei dessen Benutzung eine Zustandsänderung der CPU erfolgt. Erfindungsgemäß wird in der Decodierstufe mindestens eine zufällig ausgewählte Codesequenz als Platzhalter-Code oder Füllsel eingefügt. Dieses Verfahren ist im Prinzip für beliebige Pipelines anwendbar, die insbesondere zusätzlich zu den als Beispiel angegebenen Stufen über weitere Stufen verfügen können, und wird anhand der beigefügten Figuren näher erläutert.

[0006] Die **Fig.** 1 zeigt ein Diagramm der beschriebenen Pipeline.

[0007] Die **Fig.** 2 zeigt ein Schema für das Vorgehen beim Einfügen der Codesequenzen.

[0008] In der **Fig.** 1 ist ein Ablaufdiagramm dargestellt, das den Programmablauf von der Ladestufe **1** über die Decodierstufe **2** in die Ausführungsstufe **3** und von dort in die Rückspeicherstufe **4** einer als Beispiel dargestellten Pipeline zeigt. Die Rückspeicherstufe **4** besitzt hier mindestens ein erstes Register **41** als Scratch-Register und ein zweites Register **42** als Writeback-Register. Das Scratch-Register ist ein Register, bei dessen Benutzung keine Zustandsänderung der CPU erfolgt, während bei der Benutzung des Writeback-Registers eine Zustandsänderung der CPU erfolgt. Zur Erhöhung der Sicherheit der CPU wird von der Decodierstufe **2** eine Codesequenz, und zwar im Prinzip eine beliebige Codesequenz, in den Programmcode, der in der Pipeline übermittelt wird, eingeschleust. Es ist auch möglich, an mehreren Stellen des Programmcodes eine jeweilige zusätzliche Codesequenz als Platzhalter oder Füllsel (dummy code sequence) einzufügen. Das ist in der **Fig.** 2 im Schema dargestellt.

[0009] Die **Fig.** 2 zeigt im Schema eine Codesequenz **5** eines beliebigen Programms. In dieser Codesequenz **5** werden zufällig ausgewählte Codesequenzen **6** (Dummy-Sequenzen) an verschiedenen vorgegebenen oder ebenfalls zufällig ausgewählten Stellen eingefügt, so dass sich die erweiterte Codesequenz **50** ergibt. Die eingefügten Codesequenzen können zum Beispiel aus einem Speicher, insbesondere aus einem ROM, ausgelesen werden.

[0010] Die einzelnen Befehle zum Einfügen von Codesequenz können beispielsweise durch den Abruf von Adressen, die ein Zufallszahlengenerator erzeugt, generiert werden. Die einzufügenden Codesequenzen werden aus dem Speicher ausgelesen und an den Decoder in zufälliger Länge und Reihenfolge übermittelt. Der Decoder schleust den Code dieser Dummy-Codesequenzen in den laufenden Programmcode (Codestream) ein. Auch die Adressen, an denen der zufällig ausgewählte Code in den Programmcode eingeschleust wird, können mit einer an sich bekannten Zufallsmethode ermittelt werden.

[0011] Durch die zufallsbedingt eingefügte Codesequenz oder die mehreren zufällig ausgewählten und eingefügten Codesequenzen, die nur als Platzhalter oder Füllsel fungieren, wird keine Zustandsänderung der CPU hervorgerufen. Ein wesentlicher Vorteil dieses Verfahrens ist dabei, dass sich die Ausführungszeit des eigentlichen Programmcodes bei jedem Durchlauf desselben Programms gegenüber den vorhergehenden Durchläufen beliebig verändern lässt und dadurch ein Angriffsversuch, welchem statistische Auswertungen zugrunde liegen (wie zum Beispiel der eingangs erwähnten DPA), wesentlich erschwert ist.

Patentansprüche

1. Verfahren zur Erhöhung der Sicherheit einer CPU, bei dem eine Pipeline aus mindestens einer Decodierstufe (**2**) und einer Rückspeicherstufe (**4**) mit mindestens einem ersten Register (**41**), bei dessen Benutzung keine Zustandsänderung der CPU erfolgt, und mit mindestens einem zweiten Register (**42**), bei dessen Benutzung eine Zustandsänderung der CPU erfolgt, eingesetzt wird, **dadurch gekennzeichnet**, dass in der Decodierstufe (**2**) mindestens eine zufällig ausgewählte Codesequenz als Platzhalter-Code oder Füllsel eingefügt wird, die keine Zustandsänderung der CPU bewirkt.
2. Verfahren nach Anspruch 1, bei dem eine oder mehrere zufällig ausgewählte Codesequenzen aus einem Speicher anhand einer bzw. mehrerer zufällig ermittelter Speicheradressen ausgelesen werden.
3. Verfahren nach Anspruch 2, bei dem als Speicher ein ROM verwendet wird.
4. Verfahren nach einem der Ansprüche 1 bis 3, bei dem zusätzliche Mittel vorhanden sind, die dafür vorgesehen sind, sicherzustellen, dass bei jedem Durchlauf eines bestimmten Programms eine als Platzhalter-Code oder Füllsel verwendete und zufällig ausgewählte Codesequenz derart ausgewählt wird, dass eine jeweils von vorhergehenden Programmdurchläufen verschiedene Ausführungsdauer des Programms bewirkt wird.

Hierzu 1 Seite(n) Zeichnungen

Nummer: DE 101 01 956 A1
Int. Cl.[7]: G 06 F 12/14
Offenlegungstag: 25. Juli 2002

FIG 1

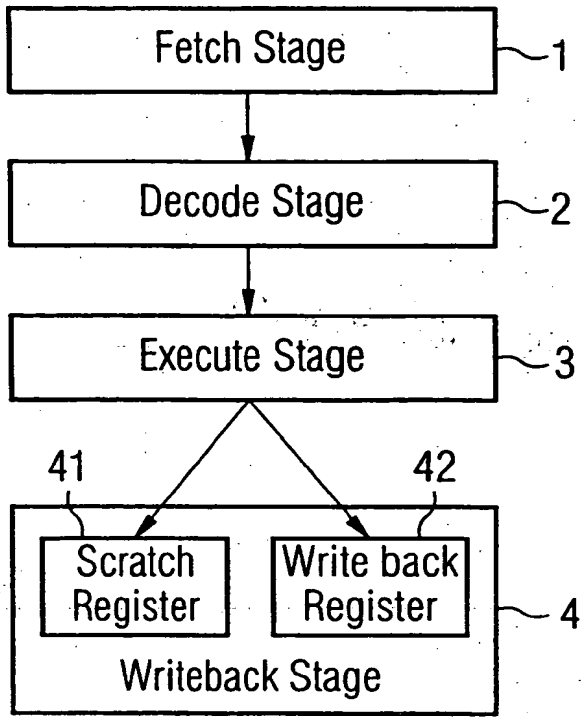
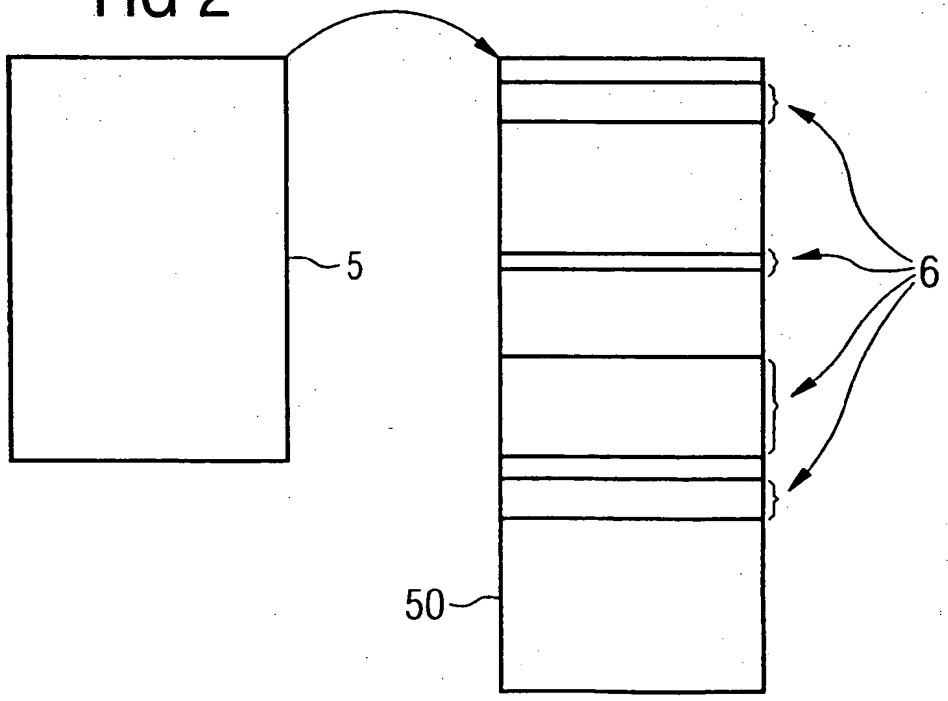


FIG 2



- Leerseite -